-- select \* from Authors as A,Books as B

-- where A.Id=B.Id\_Author

-- --SELECT \* FROM Authors AS A

-- --WHERE EXISTS(SELECT \* FROM Books AS B

-- -- WHERE B.Id\_Author=A.Id

-- --)

--IF(EXISTS(SELECT \* FROM Books AS B WHERE B.[Name]='Apple'))

--BEGIN

--PRINT 'We have apple books'

--END

--ELSE

--BEGIN

--SELECT \* FROM Books

--PRINT 'NO BOOKS like Apple'

--END

--SELECT \* FROM Students AS S

--WHERE S.Id=ANY(SELECT SC.Id\_Student FROM S\_Cards AS SC)

--SELECT \* FROM Students AS S

--WHERE S.Id=SOME(SELECT SC.Id\_Student FROM S\_Cards AS SC)

--SELECT B.[Name],B.Pages FROM Books AS B

--WHERE B.Pages < ALL(SELECT Books.Pages FROM Books,Press AS P

-- WHERE Books.Id\_Press=P.Id AND P.Id=2)

--Combinations

--JOINS

--SELECT \*

--FROM Books AS B

--CROSS JOIN Authors AS A

--WHERE B.Id\_Author=A.Id

--SELECT \*

--FROM Books AS B

--INNER JOIN Authors AS A

--ON B.Id\_Author=A.Id

--SELECT \*

--FROM Authors AS A

--INNER JOIN Books AS B

--ON B.Id\_Author=A.Id

--SELECT \*

--FROM Books AS B

-- INNER JOIN Authors AS A

-- ON B.Id\_Author=A.Id AND

-- B.Pages>(SELECT AVG(B.Pages) FROM Books AS B)

--SELECT S.FirstName,SC.DateOut

--FROM Students AS S

-- INNER JOIN S\_Cards AS SC

-- ON S.Id=SC.Id\_Student

SELECT A.Firstname ,B.[Name] AS BookName,C.[Name] AS Category

FROM Authors AS A

INNER JOIN Books AS B

ON A.Id=B.Id\_Author

INNER JOIN Categories AS C

ON C.Id=B.Id\_Category

WHERE B.Pages>150

ORDER BY A.Firstname